



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2018-0048232
(43) 공개일자 2018년05월10일

- (51) 국제특허분류(Int. Cl.)
H04L 12/741 (2013.01) H04L 12/803 (2013.01)
H04L 12/931 (2013.01) H04L 12/937 (2013.01)
(52) CPC특허분류
H04L 45/54 (2013.01)
H04L 47/125 (2013.01)
(21) 출원번호 10-2017-0028877
(22) 출원일자 2017년03월07일
심사청구일자 2017년03월07일
(30) 우선권주장
1020160144863 2016년11월02일 대한민국(KR)

- (71) 출원인
경희대학교 산학협력단
경기도 용인시 기흥구 덕영대로 1732 (서천동, 경희대학교 국제캠퍼스내)
(72) 발명자
홍충선
경기도 용인시 수지구 상현로 30-10 상현마을 성원상떼빌 233-101 (상현동, 상현마을성원상떼빌아파트)
김도현
경기도 용인시 기흥구 덕영대로 1732, 경희대학교 국제캠퍼스 전자정보대학 352호 (서천동)
(74) 대리인
김홍석, 김등용

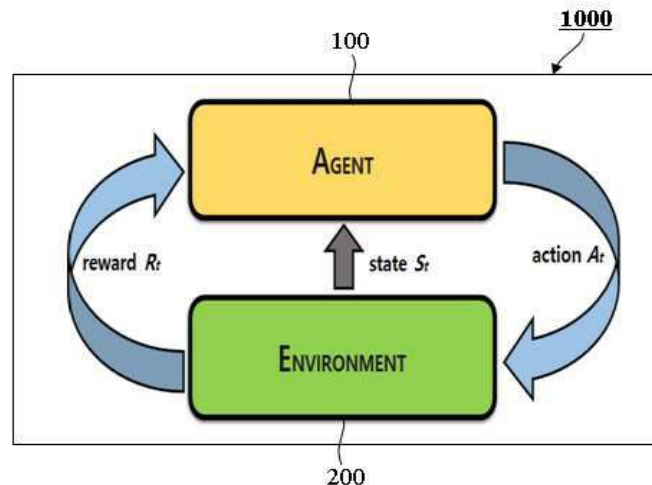
전체 청구항 수 : 총 16 항

(54) 발명의 명칭 소프트웨어 정의 네트워크에서 데이터 평면 애플리케이션을 활용하여 전송 경로를 설정하는 장치 및 방법

(57) 요약

본 발명에 따른 소프트웨어 정의 네트워크에서 데이터 평면 애플리케이션을 활용하여 전송 경로를 설정하는 전송 경로 설정 장치가 개시된다. 상기 전송 경로 설정 장치는 출발지-목적지(Src-Dst) 간 각 전송 경로에 대한 우선순위에 관한 플로우 테이블(Flow Table)을 설정하도록 구성되는 컨트롤러를 포함한다. 또한, 상기 전송 경로 설정 장치는 상기 플로우 테이블을 기반으로 해당 상태(State)마다 보상값(Reward)이 가장 큰 액션(Action)을 수행하도록 구성되는 어플리케이션 모듈을 더 포함할 수 있다. 따라서, 따라서, 본 발명에 따르면, 소프트웨어 정의 네트워크에서, 컨트롤러에 집중되는 부하를 감소시킬 수 있다.

대표도 - 도3



(52) CPC특허분류

H04L 49/253 (2013.01)

H04L 49/35 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호 IITP-2016-(H8501-16-1015)

부처명 미래창조과학부

연구관리전문기관 정보통신기술진흥센터(IITP)

연구사업명 대학ICT연구센터육성지원사업

연구과제명 실시간 모바일 클라우드 서비스 플랫폼 연구개발

기 여 율 1/2

주관기관 경희대학교 산학협력단

연구기간 2016.01.01 ~ 2016.12.31

이 발명을 지원한 국가연구개발사업

과제고유번호 R6812-16-0001

부처명 미래창조과학부

연구관리전문기관 정보통신기술진흥센터

연구사업명 Grand ICT연구센터 지원사업

연구과제명 라이프 컴패니온쉽 경험을 위한 지능형 인터랙션 융합 연구

기 여 율 1/2

주관기관 경희대학교 산학협력단

연구기간 2016.01.01 ~ 2016.12.31

명세서

청구범위

청구항 1

소프트웨어 정의 네트워크에서 데이터 평면 애플리케이션을 활용하여 전송 경로를 설정하는 장치에 있어서,
출발지-목적지(Src-Dst) 간 각 전송 경로에 대한 우선순위에 관한 플로우 테이블(Flow Table)을 설정하도록 구성되는 컨트롤러; 및

하나의 입력포트(In_port)에 대하여 전송 방향 포트(Output port)가 다수 개일 경우, 상기 플로우 테이블을 기반으로 해당 상태(State)마다 보상값(Reward)이 가장 큰 액션(Action)을 수행하도록 구성되는 어플리케이션 모듈을 포함하는, 전송 경로 설정 장치.

청구항 2

제1항에 있어서,

상기 컨트롤러는,

패킷이 호스트로부터 오픈플로우 스위치에 유입될 때, 상기 오픈플로우 스위치로부터 상기 패킷을 전송하기 위한 상기 플로우 테이블에 대한 요청 메시지(Packet_IN Message)를 수신하고,

상기 요청 메시지를 전달받아 상기 패킷을 전송할 수 있는 최단 거리 다중 경로를 선정하고,

상기 오픈플로우 스위치는,

각 플로우 테이블에서 상기 전송 방향 포트들을 활용하여 스위치의 현재 상태(State)를 결정하고,

상기 어플리케이션 모듈은,

보상값 계산모듈(R-Calculator)을 통해 상기 플로우 테이블에서 동일한 입력 포트에 대한 전송 방향 포트 정보를 토대로 보상값 테이블(Reward Table)을 생성하는, 전송 경로 설정 장치.

청구항 3

제1항에 있어서,

상기 어플리케이션 모듈은,

상기 입력 포트(In_port)를 제외한 나머지 포트들의 송신/수신 바이트들의 수의 증가 여부와 연관된 상태를 모니터링하도록 구성되는 상태 모니터(S-Monitor) 모듈; 및

상기 송신/수신 바이트들에 대하여 각각의 레이트를 계산하여, 상기 레이트의 증가 여부와 연관된 상태를 모니터링하도록 구성되는 보상값 계산모듈(R-Calculator); 및

상기 송신/수신 바이트들의 수의 증가 여부 및 상기 각각의 레이트의 증가 여부와 연관된 상태를 전달받아 해당 상태(State)마다 보상값이 가장 큰 액션(Action)을 수행하도록 구성되는 액션 컨덕터 모듈(Action conductor)을 포함하는, 전송 경로 설정 장치.

청구항 4

제3항에 있어서,

상기 상태 모니터 모듈은,

시간 $t+1$ 에서 송신/수신 바이트들의 수가 시간 t 에서의 송신/수신 바이트들의 수와 동일하면 상기 해당 상태를 0으로 정의하고,

상기 시간 $t+1$ 에서 송신/수신 바이트들의 수가 상기 시간 t 에서의 송신/수신 바이트들의 수 보다 증가하면, 상기 해당 상태를 1로 정의하여, 상기 포트들의 상태를 모니터링하는, 전송 경로 설정 장치.

청구항 5

제3항에 있어서,

상기 액션 컨덕터 모듈은,

상기 보상값에 대한 보상값 테이블 $Q(s,a)$ 에서, 패킷 유입이 검출되면 상태(s)로부터 제1액션(a)을 임의로 선택하고,

상기 제1액션(a)을 수행하여, 상기 보상값(r)을 관측하고,

상기 상태(s)로부터 제2액션(a')을 선택하여 상기 보상값 테이블 $Q(s,a)$ 를 $Q(s, a')$ 으로 업데이트하고,

상기 보상값 테이블 중 상기 상태(s)와 연관된 보상값 테이블(reward table)이 완성(complete)될 때까지 상기 보상값 관측 및 상기 보상값 테이블 업데이트를 반복하고,

상기 입력포트에 대하여 상기 전송 방향 포트가 하나만 존재한다면, 해당 플로우 엔트리에 대한 우선순위를 높여 업데이트를 수행하고,

상기 반복된 보상값 관측을 통해 상기 보상값이 가장 큰 액션을 선택하여, 경로 결정(Patch decision)이 수행되는, 전송 경로 설정 장치.

청구항 6

제3항에 있어서,

상기 보상값(Reward)은

상기 액션 컨덕터 모듈(Action conductor)이 상기 액션을 수행하였을 때 해당하는 포트에서 측정되는 전송 지연(Transmission Delay) 값의 역수와 최대 대역폭의 값에 따라 달라지는 가중치(weight)의 곱으로 계산되는 것을 특징으로 하고,

상기 액션 컨덕터 모듈은,

상기 플로우 테이블 내의 각 플로우 엔트리의 입력 포트(In port) 정보를 추출하고, 상기 입력 포트에 대응하는 적어도 하나의 출력 포트(output port)의 가용 대역폭을 기반으로 상기 보상값 계산모듈로부터 상기 보상값을 전달받아 최종 액션을 결정하는 것을 특징으로 하는, 전송 경로 설정 장치.

청구항 7

제1항에 있어서,

상기 플로우 테이블은,

복수의 플로우들을 서로 구분하도록 구성되는 플로우 엔트리;

상기 플로우 엔트리에 대하여, 상기 각 전송 경로에 대한 우선순위, 입력 포트(in_port), 상기 출발지 및 상기 목적지의 MAC 주소와 IP 주소를 포함하는 매치 필드(Match Field); 및

상기 플로우 엔트리에 대하여, 상기 입력 포트(in_port)에 대응하는 출력 포트(output port)를 포함하는 액션 필드를 포함하는, 전송 경로 설정 장치.

청구항 8

소프트웨어 정의 네트워크에서 데이터 평면 애플리케이션을 활용하여 전송 경로를 설정하는 장치에 있어서,

입력 포트(In_port)를 제외한 나머지 포트들의 송신/수신 바이트들의 수의 증가 여부와 연관된 상태를 모니터링하도록 구성되는 상태 모니터 모듈; 및

하나의 입력포트(In_port)에 대하여 전송 방향 포트(Output port)가 다수 개일 경우, 송신/수신 바이트들의 수의 증가 여부와 연관된 상태를 전달받아 해당 상태(State)마다 보상값(Reward)이 가장 큰 액션(Action)을 수행하도록 구성되는 액션 컨덕터 (Action Conductor) 모듈을 포함하는, 전송 경로 설정 장치.

청구항 9

제8항에 있어서,

상기 송신/수신 바이트들에 대하여 각각의 레이트를 계산하여, 상기 레이트의 증가 여부와 연관된 상태를 모니터링하도록 구성되는 보상값 계산모듈(R-Calculator)을 더 포함하고,

상기 액션 컨덕터 모듈은, 상기 송신/수신 바이트들의 수의 증가 여부 및 상기 각각의 레이트의 증가 여부와 연관된 상태를 전달받아 해당 상태(State)마다 보상값이 가장 큰 액션(Action)을 수행하도록 구성되는, 전송 경로 설정 장치.

청구항 10

제8항에 있어서,

상기 상태 모니터 모듈은,

시간 $t+1$ 에서 송신/수신 바이트들의 수가 시간 t 에서의 송신/수신 바이트들의 수와 동일하면 상기 해당 상태를 0으로 정의하고,

상기 시간 $t+1$ 에서 송신/수신 바이트들의 수가 상기 시간 t 에서의 송신/수신 바이트들의 수보다 증가하면, 상기 해당 상태를 1로 정의하여, 상기 포트들의 상태를 모니터링하는, 전송 경로 설정 장치.

청구항 11

제10항에 있어서,

상기 액션 컨덕터 모듈은,

상기 포트들의 사용 유무와 상기 수신된 플로우 테이블에 기반하여, 상기 해당 상태(State)마다 상기 보상값(Reward)이 가장 큰 액션(Action)을 수행하도록 구성되고,

상기 포트들의 사용 유무에 의해 상기 상태가 정의되어, 상기 상태에 대한 액션으로부터의 보상값은 상기 포트들의 사용 유무에 따라 달라지는 것을 특징으로 하는, 전송 경로 설정 장치.

청구항 12

소프트웨어 정의 네트워크에서 데이터 평면 애플리케이션을 활용하여 전송 경로를 설정하는 방법에 있어서,

입력 포트(In_port)를 제외한 나머지 포트들의 송신/수신 바이트들의 수의 증가 여부와 연관된 상태를 모니터링하는 상태 모니터링 단계;

상기 송신/수신 바이트들에 대하여 각각의 레이트를 계산하여, 상기 레이트의 증가 여부와 연관된 상태를 모니터링하는 레이트 모니터링 단계; 및

상기 송신/수신 바이트들의 수의 증가 여부 및 상기 각각의 레이트의 증가 여부와 연관된 상태를 전달받아 해당 상태(State)마다 보상값이 가장 큰 액션(Action)을 수행하는 액션 수행 단계를 포함하는, 전송 경로 설정 방법.

청구항 13

제12항에 있어서,

상기 상태 모니터링 단계는,

시간 $t+1$ 에서 송신/수신 바이트들의 수가 시간 t 에서의 송신/수신 바이트들의 수와 동일하면 상기 해당 상태를 0으로 정의하고,

상기 시간 $t+1$ 에서 송신/수신 바이트들의 수가 상기 시간 t 에서의 송신/수신 바이트들의 수보다 증가하면, 상기 해당 상태를 1로 정의하여, 상기 포트들의 상태를 모니터링하는, 전송 경로 설정 방법.

청구항 14

제12항에 있어서,

상기 액션 수행 단계는,

상기 보상값에 대한 보상값 테이블 $Q(s,a)$ 에서, 패킷 유입이 검출되면 상태(s)로부터 제1액션(a)을 임의로 선택하는 제1액션 선택 단계;

상기 제1액션(a)을 수행하여, 상기 보상값(r)을 관측하는 보상값 관측 단계; 및

상기 상태(s)로부터 제2액션(a')을 선택하여 상기 보상값 테이블 $Q(s,a)$ 를 $Q(s, a')$ 으로 업데이트하는 제2액션 선택/보상값 테이블 업데이트 단계를 포함하는, 전송 경로 설정 방법.

청구항 15

제14항에 있어서,

상기 액션 수행 단계는,

상기 보상값 테이블 중 상기 상태(s)와 연관된 보상값 테이블(reward table)의 완성(complete) 여부를 판단하는 보상값 테이블 완성 여부 판단 단계; 및

상기 보상값 테이블이 완성될 때까지 상기 보상값 관측 단계 및 상기 제2액션 선택/보상값 테이블 업데이트 단계를 반복하고, 상기 반복된 보상값 관측을 통해 상기 보상값이 가장 큰 액션을 선택하여, 경로 결정(Patch decision)을 수행하는 경로 결정 단계를 더 포함하는, 전송 경로 설정 방법.

청구항 16

제12항 내지 제15항 중 어느 한 항의 전송 경로 설정 방법을 수행하는 프로그램이 기록된 컴퓨터로 판독가능한 기록매체.

발명의 설명

기술 분야

[0001] 본 발명은 소프트웨어 정의 네트워크에서 전송 경로를 설정하는 장치 및 방법에 관한 것이다. 보다 상세하게는, 소프트웨어 정의 네트워크에서 데이터 평면 애플리케이션을 활용하여 전송 경로를 설정하는 장치 및 방법에 관한 것이다.

배경 기술

[0002] 소프트웨어 정의 네트워크는 기존 라우터 등의 네트워크 장비에 공존하는 제어 평면과 데이터 평면을 분리하여 다양한 네트워크의 기능들을 소프트웨어적으로 구현하고 적용시킬 수 있는 네트워킹 기술이다. 전통적인 네트워크 환경에서는 특정한 기능, 예를 들어 방화벽, Deep Packet Inspection, Traffic Monitor 등의 기능을 적용시키려 할 때, 해당 기능을 수행하는 고가의 장비를 구입하고 설치해야 하기 때문에 비용적인 측면과 유지보수의 측면에서 많은 부담이 되었다.

[0003] 하지만, 소프트웨어 정의 네트워크 환경에서는 앞서 언급했듯 프로그래밍을 통해 해당 기능을 구현/개발할 수 있기 때문에 최근 다양한 네트워크 환경(Internet of Things, Data Center Network)에 적용되고 그에 알맞은 관리 기법 또한 연구/개발되고 있다. 소프트웨어 정의 네트워크의 전체적인 구조는 어플리케이션 계층(Application layer), 제어 평면(Control Plane), 데이터 평면(Data plane)으로 나누어질 수 있다. 이와 관련하여, 도 1과 도 2는 본 발명과 관련된 기존 소프트웨어 정의 네트워크의 기본적인 구조를 나타낸다.

[0004] 하지만, 이러한 기존 소프트웨어 정의 네트워크의 경우, 네트워크 운용 및 관리에 대한 많은 기능들이 오픈플로우 스위치 상에 위치하게 된다. 또한, 소프트웨어 정의 네트워크와 연동되는 특정 분야의 필요 기능들도 Application Layer에 위치하여 컨트롤러와 연계를 통해 기능을 수행한다. 이러한 이유로 많은 노드들이 포함된 대규모 네트워크를 컨트롤러만으로 운용/관리하기에는 컨트롤러 과부하가 발생하기 쉽고 더 나아가 네트워크가 효율적으로 운용되지 못하는 문제가 발생할 수 있다.

발명의 내용

해결하려는 과제

- [0005] 따라서, 본 발명에서 해결하고자 하는 과제는, 소프트웨어 정의 네트워크에서, 컨트롤러에 집중되는 부하를 감소시키는 데에 그 목적이 있다.
- [0006] 또한, 본 발명에서 해결하고자 하는 과제는, 소프트웨어 정의 네트워크에서, 컨트롤러에 집중되는 부하에 따라 사용자 데이터의 전송 지연이 증가하는 것을 감소시키는 데에 그 목적이 있다.

과제의 해결 수단

- [0007] 상기와 같은 과제를 해결하기 위한 본 발명에 따른 소프트웨어 정의 네트워크에서 데이터 평면 애플리케이션을 활용하여 전송 경로를 설정하는 전송 경로 설정 장치가 개시된다. 상기 전송 경로 설정 장치는 하나의 입력포트(In_port)에 대하여 전송 방향 포트(Output port)가 다수 개일 경우, 출발지-목적지(Src-Dst) 간 각 전송 경로에 대한 우선순위에 관한 플로우 테이블(Flow Table)을 설정하도록 구성되는 컨트롤러를 포함한다. 또한, 상기 전송 경로 설정 장치는 상기 플로우 테이블을 기반으로 해당 상태(State)마다 보상값(Reward)이 가장 큰 액션(Action)을 수행하도록 구성되는 어플리케이션 모듈을 더 포함할 수 있다. 따라서, 본 발명에 따르면, 소프트웨어 정의 네트워크에서, 컨트롤러에 집중되는 부하를 감소시킬 수 있다.
- [0008] 일 실시예에 따르면, 상기 컨트롤러는, 패킷이 호스트로부터 오픈플로우 스위치에 유입될 때, 상기 오픈플로우 스위치로부터 상기 패킷을 전송하기 위한 상기 플로우 테이블에 대한 요청 메시지(Packet_IN Message)를 수신할 수 있다. 또한, 상기 컨트롤러는, 상기 요청 메시지를 전달받아 상기 패킷을 전송할 수 있는 최단 거리 다중 경로를 선정할 수 있다.
- [0009] 일 실시예에 따르면, 상기 오픈플로우 스위치는, 각 플로우 테이블에서 상기 전송 방향 포트들을 활용하여 스위치의 현재 상태(State)를 결정할 수 있다.
- [0010] 일 실시예에 따르면, 상기 어플리케이션 모듈은, 보상값 계산모듈(R-Calculator)을 통해 상기 플로우 테이블에서 동일한 입력 포트에 대한 전송 방향 포트 정보를 토대로 보상값 테이블(Reward Table)을 생성할 수 있다.
- [0011] 일 실시예에 따르면, 상기 플로우 테이블은, 복수의 플로우들을 서로 구분하도록 구성되는 플로우 엔트리; 상기 플로우 엔트리에 대하여, 상기 각 전송 경로에 대한 우선순위, 입력 포트(in_port), 상기 출발지 및 상기 목적지의 MAC 주소와 IP 주소를 포함하는 매치 필드(Match Field); 및 상기 플로우 엔트리에 대하여, 상기 입력 포트(in_port)에 대응하는 출력 포트(output port)를 포함하는 액션 필드를 포함할 수 있다.
- [0012] 일 실시예에 따르면, 상기 어플리케이션 모듈은, 입력 포트(In_port)를 제외한 나머지 포트들의 송신/수신 바이트들의 수의 증가 여부와 연관된 상태를 모니터링하도록 구성되는 상태 모니터(S-Monitor) 모듈을 포함한다. 또한, 상기 어플리케이션 모듈은, 상기 송신/수신 바이트들에 대하여 각각의 레이트를 계산하여, 상기 레이트의 증가 여부와 연관된 상태를 모니터링하도록 구성되는 보상값 계산모듈(R-Calculator)을 더 포함할 수 있다. 또한, 상기 어플리케이션 모듈은, 상기 송신/수신 바이트들의 수의 증가 여부 및 상기 각각의 레이트의 증가 여부와 연관된 상태를 전달받아 해당 상태(State)마다 보상값이 가장 큰 액션(Action)을 수행하도록 구성되는 액션 컨덕터 모듈(Action conductor)을 더 포함할 수 있다.
- [0013] 일 실시예에 따르면, 상기 상태 모니터 모듈은, 시간 t+1에서 송신/수신 바이트들의 수가 시간 t에서의 송신/수신 바이트들의 수와 동일하면 상기 해당 상태를 0으로 정의하고, 상기 시간 t+1에서 송신/수신 바이트들의 수가 상기 시간 t에서의 송신/수신 바이트들의 수 보다 증가하면, 상기 해당 상태를 1로 정의하여, 상기 포트들의 상태를 모니터링할 수 있다.
- [0014] 일 실시예에 따르면, 상기 액션 컨덕터 모듈은, 상기 보상값에 대한 보상값 테이블 $Q(s,a)$ 에서, 패킷 유입이 검출되면 상태(s)로부터 제1액션(a)을 임의로 선택하고, 상기 제1액션(a)을 수행하여, 상기 보상값(r)을 관측하고, 상기 상태(s)로부터 제2액션(a')을 선택하여 상기 보상값 테이블 $Q(s,a)$ 를 $Q(s, a')$ 으로 업데이트할 수 있다. 또한, 상기 액션 컨덕터 모듈은, 상기 보상값 테이블 중 상기 상태(s)와 연관된 보상값 테이블(reward table)이 완성(complete)될 때까지 상기 보상값 관측 및 상기 보상값 테이블 업데이트를 반복하고, 상기 반복된 보상값 관측을 통해 상기 보상값이 가장 큰 액션을 선택하여, 경로 결정(Patch decision)이 수행될 수 있다. 이때, 상기 입력포트에 대하여 상기 전송 방향 포트가 하나만 존재한다면, 해당 플로우 엔트리에 대한 우선순위를 높여 업데이트를 수행할 수 있다.
- [0015] 일 실시예에 따르면, 상기 보상값(Reward)은 상기 액션 컨덕터 모듈(Action conductor)이 상기 액션을 수행하였

을 때 해당하는 포트에서 측정되는 전송 지연(Transmission Delay) 값의 역수와 최대 대역폭의 값에 따라 달라지는 가중치(weight)의 곱으로 계산될 수 있다.

[0016] 일 실시예에 따르면, 상기 액션 컨덕터 모듈은, 상기 플로우 테이블 내의 각 플로우 엔트리의 입력 포트(In port) 정보를 추출하고, 상기 입력 포트에 대응하는 적어도 하나의 출력 포트(output port)의 가용 대역폭을 기반으로 상기 보상값 계산모듈로부터 상기 보상값을 전달받아 최종 액션을 결정할 수 있다.

[0017] 또한, 본 발명의 다른 양상에 따른 소프트웨어 정의 네트워크에서 데이터 평면 애플리케이션을 활용하여 전송 경로를 설정하는 장치가 개시된다. 상기 전송 경로 설정 장치는, 입력 포트(In_port)를 제외한 나머지 포트들의 송신/수신 바이트들의 수의 증가 여부와 연관된 상태를 모니터링하도록 구성되는 상태 모니터 모듈을 포함한다. 또한, 상기 전송 경로 설정 장치는, 송신/수신 바이트들의 수의 증가 여부와 연관된 상태를 전달받아 해당 상태(State)마다 보상값(Reward)이 가장 큰 액션(Action)을 수행하도록 구성되는 액션 컨덕터 (Action Conductor) 모듈을 더 포함한다.

[0018] 일 실시예에 따르면, 상기 전송 경로 설정 장치는, 상기 송신/수신 바이트들에 대하여 각각의 레이트를 계산하여, 상기 레이트의 증가 여부와 연관된 상태를 모니터링하도록 구성되는 보상값 계산모듈(R-Calculator)을 더 포함할 수 있다. 이때, 상기 액션 컨덕터 모듈은, 상기 송신/수신 바이트들의 수의 증가 여부 및 상기 각각의 레이트의 증가 여부와 연관된 상태를 전달받아 해당 상태(State)마다 보상값이 가장 큰 액션(Action)을 수행하도록 구성될 수 있다.

[0019] 일 실시예에 따르면, 상기 상태 모니터 모듈은, 시간 t+1에서 송신/수신 바이트들의 수가 시간 t에서의 송신/수신 바이트들의 수와 동일하면 상기 해당 상태를 0으로 정의하고, 상기 시간 t+1에서 송신/수신 바이트들의 수가 상기 시간 t에서의 송신/수신 바이트들의 수보다 증가하면, 상기 해당 상태를 1로 정의하여, 상기 포트들의 상태를 모니터링할 수 있다.

[0020] 일 실시예에 따르면, 상기 액션 컨덕터 모듈은, 상기 포트들의 사용 유무와 상기 수신된 플로우 테이블에 기반하여, 상기 해당 상태(State)마다 상기 보상값(Reward)이 가장 큰 액션(Action)을 수행하도록 구성될 수 있다. 또한, 상기 액션 컨덕터 모듈은, 상기 포트들의 사용 유무에 의해 상기 상태가 정의되어, 상기 상태에 대한 액션으로부터의 보상값은 상기 포트들의 사용 유무에 따라 달라질 수 있다.

[0021] 또한, 본 발명의 또 다른 양상에 따른 소프트웨어 정의 네트워크에서 데이터 평면 애플리케이션을 활용하여 전송 경로를 설정하는 방법이 개시된다. 상기 전송 경로 설정 방법은, 입력 포트(In_port)를 제외한 나머지 포트들의 송신/수신 바이트들의 수의 증가 여부와 연관된 상태를 모니터링하는 상태 모니터링 단계를 포함한다. 또한, 상기 전송 경로 설정 방법은, 상기 송신/수신 바이트들에 대하여 각각의 레이트를 계산하여, 상기 레이트의 증가 여부와 연관된 상태를 모니터링하는 레이트 모니터링 단계를 더 포함한다. 또한, 상기 전송 경로 설정 방법은, 상기 송신/수신 바이트들의 수의 증가 여부 및 상기 각각의 레이트의 증가 여부와 연관된 상태를 전달받아 해당 상태(State)마다 보상값이 가장 큰 액션(Action)을 수행하는 액션 수행 단계를 더 포함한다.

[0022] 일 실시예에 따르면, 상기 상태 모니터링 단계는, 시간 t+1에서 송신/수신 바이트들의 수가 시간 t에서의 송신/수신 바이트들의 수와 동일하면 상기 해당 상태를 0으로 정의하고, 상기 시간 t+1에서 송신/수신 바이트들의 수가 상기 시간 t에서의 송신/수신 바이트들의 수보다 증가하면, 상기 해당 상태를 1로 정의하여, 상기 포트들의 상태를 모니터링할 수 있다.

[0023] 일 실시예에 따르면, 상기 액션 수행 단계는, 상기 보상값에 대한 보상값 테이블 $Q(s,a)$ 에서, 패킷 유입이 검출되면 상태(s)로부터 제1액선(a)을 임의로 선택하는 제1액선 선택 단계; 상기 제1액선(a)을 수행하여, 상기 보상값(r)을 관측하는 보상값 관측 단계; 및 상기 상태(s)로부터 제2액선(a')을 선택하여 상기 보상값 테이블 $Q(s,a)$ 를 $Q(s, a')$ 으로 업데이트하는 제2액선 선택/보상값 테이블 업데이트 단계를 포함할 수 있다.

[0024] 일 실시예에 따르면, 상기 액션 수행 단계는, 상기 보상값 테이블 중 상기 상태(s)와 연관된 보상값 테이블(reward table)의 완성(complete) 여부를 판단하는 보상값 테이블 완성 여부 판단 단계; 및 상기 보상값 테이블이 완성될 때까지 상기 보상값 관측 단계 및 상기 제2액선 선택/보상값 테이블 업데이트 단계를 반복하고, 상기 반복된 보상값 관측을 통해 상기 보상값이 가장 큰 액션을 선택하여, 경로 결정(Patch decision)을 수행하는 경로 결정 단계를 더 포함할 수 있다.

발명의 효과

[0025] 본 발명에 따른 전송 경로 설정 방법은, 소프트웨어 정의 네트워크에서 패킷전송만을 담당했던 데이터 평면에

전송 경로 갱신 및 요청 등의 기능을 Application 형태로 배치하여 처리함으로써, 컨트롤러에 집중되는 부하를 감소시킬 수 있다는 장점이 있다.

- [0026] 또한, 본 발명에 따른 전송 경로 설정 방법은, 소프트웨어 정의 네트워크에서 패킷전송만을 담당했던 데이터 평면에 전송 경로 갱신 및 요청 등의 기능을 Application 형태로 배치하여 처리함으로써, 컨트롤러에 집중되는 부하에 따라 사용자 데이터의 전송 지연이 증가하는 것을 감소시킬 수 있다는 장점이 있다.

도면의 간단한 설명

- [0027] 도 1은 본 발명과 관련된 기존 소프트웨어 정의 네트워크의 기본적인 구조를 나타낸다.
- 도 2는 본 발명에 따른 소프트웨어 정의 네트워크에서 전송 경로를 설정하는 장치의 개념도이다.
- 도 3는 본 발명에 따른 소프트웨어 정의 네트워크에서 전송 경로를 설정하는 방법과 관련하여, 강화학습의 기본 동작 절차를 나타낸 것이다.
- 도 4은 본 발명에 따른 소프트웨어 정의 네트워크에서 데이터 평면 애플리케이션을 활용하여 전송 경로를 설정하는 전송 경로 설정 장치의 구성 요소들을 나타낸다.
- 도 5는 본 발명에 따른 Application (또는 어플리케이션 모듈)이 OpenVSwitch(오픈플로우 스위치)로부터 얻을 수 있는 각 상태를 나타낸 도면이다.
- 도 6는 본 발명의 일 실시예에 따른 각각의 입력 포트에서의 보상값 테이블을 도시한다.
- 도 7은 본 발명의 일 실시예에 따라, Packet_IN 메시지를 기반으로 컨트롤러가 OpenVSwitch에 플로우 테이블을 배치한 것을 나타낸다.
- 도 8은 본 발명의 일 실시예에 따라, 전송 경로가 설정된 형태를 나타낸다.
- 도 9은 본 발명에 따른 소프트웨어 정의 네트워크에서 데이터 평면 애플리케이션을 활용하여 전송 경로를 설정하는 전송 경로 설정 방법의 흐름도를 도시한다.

발명을 실시하기 위한 구체적인 내용

- [0028] 상술한 본 발명의 특징 및 효과는 첨부된 도면과 관련한 다음의 상세한 설명을 통하여 보다 분명해 질 것이며, 그에 따라 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 본 발명의 기술적 사상을 용이하게 실시할 수 있을 것이다. 본 발명은 다양한 변경을 가할 수 있고 여러 가지 형태를 가질 수 있는바, 특정 실시 예들을 도면에 예시하고 본문에 상세하게 설명하고자 한다. 그러나 이는 본 발명을 특정한 개시형태에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다. 본 명세서에서 사용한 용어는 단지 특정한 실시 예들을 설명하기 위해 사용된 것으로, 본 발명을 한정하려는 의도가 아니다.
- [0029] 각 도면을 설명하면서 유사한 참조부호를 유사한 구성요소에 대해 사용한다.
- [0030] 제1, 제2등의 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 상기 구성요소들은 상기 용어들에 의해 한정되어서는 안 된다. 상기 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용된다.
- [0031] 예를 들어, 본 발명의 권리 범위를 벗어나지 않으면서 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다. "및/또는" 이라는 용어는 복수의 관련된 기재된 항목들의 조합 또는 복수의 관련된 기재된 항목들 중의 어느 항목을 포함한다.
- [0032] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미가 있다.
- [0033] 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 의미를 가지는 것으로 해석되어야 하며, 본 출원에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않아야 한다.
- [0034] 이하의 설명에서 사용되는 구성요소에 대한 접미사 "모듈", "블록" 및 "부"는 명세서 작성의 용이함만이 고려되어 부여되거나 혼용되는 것으로서, 그 자체로 서로 구별되는 의미 또는 역할을 갖는 것은 아니다.

- [0035] 이하, 본 발명의 바람직한 실시 예를 첨부한 도면을 참조하여 당해 분야에 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 설명한다. 하기에서 본 발명의 실시 예를 설명함에 있어, 관련된 공지의 기능 또는 공지의 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략한다.
- [0036] 이하, 본 발명에 따른 소프트웨어 정의 네트워크에서 데이터 평면 애플리케이션을 활용하여 전송 경로를 설정하는 전송 경로 설정 장치 및 방법에 대해 살펴보기로 한다. 여기서, 전송 경로 설정 장치는 소스/목적지 노드뿐만 아니라, 다수의 노드들을 포함하는 네트워크상의 임의의 노드일 수 있다. 따라서, 전송 경로 설정 장치는 사용자 단말 이외에 네트워크 기기, 라우터 중 하나일 수 있다.
- [0037] 도 2는 본 발명에 따른 소프트웨어 정의 네트워크에서 전송 경로를 설정하는 장치의 개념도이다.
- [0038] 도 3은 본 발명에 따른 소프트웨어 정의 네트워크에서 전송 경로를 설정하는 방법과 관련하여, 강화학습의 기본 동작 절차를 나타낸 것이다.
- [0039] 이러한 본 발명에 따른 소프트웨어 정의 네트워크에서 전송 경로를 설정하는 방법과 관련하여, 강화 학습(Reinforcement Learning)에 의한 전송 경로 설정이 수행될 수 있다. 따라서, 강화 학습(Reinforcement Learning)에 의한 전송 경로 설정에 대해 살펴보면 다음과 같다.
- [0040] 강화학습은 기계학습의 범주에 속하는 학습방법 중 하나로, 적용하고자 하는 환경 안에서 Agent가 현재의 State(상태)를 관측하여 수행 가능한 행동들 중 Reward(보상값)를 최대화하거나 Penalty(불이익)를 최소화하는 행동 혹은 행동순서를 선택하여 학습해 나가는 방식이다. 이는 Markov Decision Process(MDP) 모델을 기반으로 Agent가 얻을 수 있는 State와 Agent가 수행하는 Action, 그리고 Action에 따른 Reward/Penalty를 정의하여 Environment와의 상호작용을 통해 점진적인 학습절차를 거치게 된다. 한편, 도 2는 본 발명에 따른 소프트웨어 정의 네트워크에서 전송경로를 설정하는 장치의 개념도를 나타낸 것이며, 도 3은 본 발명에 따른 소프트웨어 정의 네트워크에서 전송 경로를 설정하는 방법과 관련하여, 강화학습의 기본 동작 절차를 나타낸 것이다. 도 2, 도 3을 참조하면, 상기 강화학습의 기본 동작 절차를 수행하는 전송 경로 설정 장치(1000)는 어플리케이션 모듈(100)과 오픈플로우 스위치(200)를 포함한다. 여기서, 어플리케이션 모듈(100)은 에이전트(agent)로 지칭되고, 오픈플로우 스위치(200)는 (런타임) 환경 모듈(Runtime Environment module)로 지칭될 수 있다.
- [0041] 도 3에 도시된 바와 같이, 어플리케이션 모듈(100)은 오픈플로우 스위치(200)로부터 상태(State, St)에 관한 정보를 수신하고, 이에 기반하여 특정한 액션(action, At)을 수행한다. 이러한 특정한 액션(action, At)에 기반하여, 오픈플로우 스위치(200)는 보상값(reward, Rt)을 어플리케이션 모듈(100)로 전달한다. 이러한 액션에 기반하여 보상값을 구하는 과정을 반복함으로써 최적의 보상값을 구하는 것이 강화학습의 궁극적인 목표에 해당한다.
- [0042] 이러한 강화학습과 관련하여, SARSA(State-Action-Reward-State-Action) 알고리즘은 강화학습 알고리즘 중 하나로, 각 State에서 Action을 수행한 후, 그에 따른 Reward를 받아 Q 테이블을 업데이트하는 알고리즘이다.
- [0043] 이와 관련하여, 표 1은 본 발명에 따른 SARSA 알고리즘의 기본 개념을 나타낸다.

표 1

```

Initialize  $Q(s, a), \forall s \in S, a \in \mathcal{A}(s)$ , arbitrarily, and  $Q(\text{terminal-state}, \cdot) = 0$ 
Repeat (for each episode):
  Initialize  $S$ 
  Choose  $A$  from  $S$  using policy derived from  $Q$  (e.g.,  $\epsilon$ -greedy)
  Repeat (for each step of episode):
    Take action  $A$ , observe  $R, S'$ 
    Choose  $A'$  from  $S'$  using policy derived from  $Q$  (e.g.,  $\epsilon$ -greedy)
     $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma Q(S', A') - Q(S, A)]$ 
     $S \leftarrow S'; A \leftarrow A';$ 
  until  $S$  is terminal
  
```

- [0044]
- [0045] 즉, 본 발명에 따른 특정 액션에 기반하여 보상값을 구하는 강화 학습 과정에서는, Q라는 보상값 함수(또는 보상값 테이블)를 모든 상태(s)와 액션(a)에 대하여 초기화함으로써 시작된다. 이후, 각각의 에피소드에 대하여, 상태 집합(S)을 초기화하고, Q로부터의 정책(policy)을 이용하여 상태 집합(S)으로부터 특정 액션 A를

선택한다. 이후, 특정 액션 A를 수행하여, 보상값(R)와 업데이트된 상태 집합(S')을 획득한다. 다음으로, Q로부터의 정책을 이용하여 업데이트된 상태 집합(S')로부터 업데이트된 특정 액션(A')을 선택한다. 전술된 바와 같이, 선택된 액션을 수행하고, 보상값을 구하는 과정은, 상태 집합이 종료(terminal) 상태에 이를 때까지 반복될 수 있다. 이러한, 종료 상태는 각 에피소드에 대하여 충분히 필요한 액션이 수행되었거나 획득된 보상값이 수렴되거나 또는 특정 값 이상을 갖는 경우가 이에 해당할 수 있다.

[0046] 한편, 도 4는 본 발명에 따른 소프트웨어 정의 네트워크에서 데이터 평면 애플리케이션을 활용하여 전송 경로를 설정하는 전송 경로 설정 장치의 구성 요소들을 나타낸다. 즉, 본 발명에서 제안하는 경로설정 기능을 위한 데이터 평면 Application 구조를 나타낸다. 도 2와 도4에 도시된 바와 같이 본 발명에서 전송경로를 설정하는 것은 컨트롤러(10)에서 데이터 평면상의 노드들에게 전달하는 전송경로를 위한 Flow Table을 Action으로 정의하고 각 State별로 Action에 대한 Reward를 얻음으로써 다음 State의 Action을 결정하는 방식이다.

[0047] 한편, SDN 환경에서는 오픈플로우 스위치 상에 모니터링 기능을 구현하여 모든 노드에 대한 정보를 수집하고 그에 따른 전송 경로를 설정하는 방식이 사용될 수 있다. 하지만, 컨트롤러(10)가 해야 하는 작업 또는 관리하여야 할 부분이 많을 경우, 네트워크상의 Data Plane에 노드가 많을수록 과부하가 발생할 수 있기 때문에 모니터링 기능을 Data Plane 상의 각 노드로 분산시켜 배치할 수 있다.

[0048] 도 4에 도시된 바와 같이, 전송 경로 설정 장치(1000)는 어플리케이션 모듈(100)과 오픈플로우 스위치(200)를 포함한다. 여기서, 어플리케이션 모듈(100)과 오픈플로우 스위치(200)는 각각 DP Application과 Environment로 지칭될 수 있다. 특히, Environment의 일 예로 런타임 환경(Runtime Environment)이 이에 해당할 수 있다.

[0049] 또한, 어플리케이션 모듈(100)은 상태 모니터(S-Monitor)모듈(110), 보상값 계산모듈(R-Calculator) 모듈(120)과 액션 컨덕터 (Action conductor)모듈(130)을 포함한다. 여기서, 상태 모니터 모듈(110), 보상값 계산모듈(120)과 액션 컨덕터 모듈(130)은 각각 S Monitor, R Calculator와 Action conductor로 지칭될 수 있다.

[0050] 또한, 오픈플로우 스위치(200)는 포트 상태 제공 모듈(210) 및 플로우 테이블 모듈(220)을 포함한다. 여기서, 포트 상태 제공 모듈(210) 및 플로우 테이블 모듈(220)은 각각 Physical port status (providing module) 및 Flow Table (module)로 지칭될 수 있다. 한편, 플로우 테이블 모듈(220)은 플로우 테이블을 포함하는 오픈플로우 스위치로도 지칭될 수 있다. 또한, Physical port status (providing module)에 해당하는 포트 상태 제공 모듈(210)의 일 예로, 네트워크 인터페이스 카드(NIC: Network Interface Card)가 이에 해당될 수 있다.

[0051] 한편, 도 4의 좌측에 도시된 바와 같이, 어플리케이션 모듈(100)과 오픈플로우 스위치(200)는 각각 Applicaiton 및 OVS(Open VSwitch)로 지칭될 수 있다. OpenVSwitch의 State는 SR Monitor 모듈을 통한 포트 모니터링에 의해 결정된다. SR Monitor 모듈에서는 Host와 연결된 Port를 제외한 나머지 포트들의 시간 t+1에서 Tx bytes, Rx bytes와 시간 t에서의 Tx bytes, Rx bytes가 동일할 경우 0(미사용중), 증가했을 경우 1(사용중)로 나타내며 Output Port들이 얻는 0 또는 1값의 집합을 State로 정의한 후, 요청이 있을 시 모니터링 하여 액션 컨덕터 (Action conductor) 모듈(130)로 전달하게 된다. 또한, 상기 보상값(Reward)은 액션 컨덕터 모듈(130)이 해당 액션(Action)을 수행하였을 때 해당하는 Port에서 측정되는 전송 지연(Transmission Delay) 값의 역수로 계산된다. 또한, 상기 보상값(Reward)은 상기 액션 컨덕터 모듈(Action conductor)이 상기 액션을 수행하였을 때 해당하는 포트에서 측정되는 전송 지연(Transmission Delay)값의 역수와 최대 대역폭의 값에 따라 달라지는 가중치(weight)의 곱으로 계산될 수 있다. 이에 관해서는 아래에서 자세히 살펴보기로 한다.

[0052] 전술된 내용과 관련하여, 도 4에서의 어플리케이션 모듈(100)과 오픈플로우 스위치(200)의 관점에서, 상세히 살펴보면 다음과 같다.

[0053] 먼저, 도 2, 도 4와 관련하여, 패킷이 전송되는 전체적인 전송경로 설정 과정을 살펴보면 다음의 순서와 같다. 한편, 이러한 전송경로 설정 순서는 다음의 순서에 한정되는 것이 아니라, 본 발명의 범위 내에서 변경 가능하다.

[0054] 1) 패킷이 호스트로부터 오픈플로우 스위치(200)에 유입될 때, 오픈플로우 스위치(200)는 SDN 컨트롤러(10)에게 해당 패킷을 전송하기 위한 다중 경로의 플로우 테이블을 요청한다.

[0055] 2) 요청 메시지(Packet_IN Message)를 전달받은 SDN 컨트롤러(10)는 패킷을 전송할 수 있는 최단 거리 다중 경로를 선정한다. 이때, 최단 거리 다중 경로는 다중 경로에 대한 플로우 테이블에서 단순히 Hop Count를 활용하여 선정된다. 이후에, SDN 컨트롤러(10)는 최단 거리 다중 경로를 각각의 오픈플로우 스위치(200)로 전달하게 된다.

- [0056] 3) 플로우 테이블을 전달받은 각 오픈플로우 스위치(200)는 각 플로우 테이블에서 전송 방향(Output port)의 포트들을 활용하여 스위치의 현재 상태(State)를 결정한다. 이러한, 스위치의 현재 상태(State)는 상태 모니터(State Monitor)모듈(110)을 통해 모니터링되는 상태를 이용하여 이루어진다.
- [0057] 4) 또한, 플로우 테이블에서 동일한 입력 포트에 대한 전송 방향 포트 정보를 토대로 보상값 테이블(Reward Table)을 생성한다. 이러한 보상값 테이블은 보상값 계산모듈(R-Calculator) 모듈(120)을 통해 계산되는 보상값에 기반하여 생성된다. 한편, 이러한 보상값 테이블의 일 예시는 추후 기술될 도 6에 도시되는 보상값 테이블일 수 있으며, 이에 대해서는 아래에서 자세히 살펴보기로 한다.
- [0058] 5) 이때, 하나의 입력포트에 대하여 전송 방향 포트가 다수 개일 경우, 각 전송 방향 포트의 Reward 값 중 가장 큰 Reward 값을 갖는 포트를 선택하여 플로우 테이블에 반영한다. 여기서, 가장 큰 Reward 값을 갖는 포트를 선택하는 것은 해당하는 플로우 엔트리의 우선순위를 높여 해당 전송 방향 포트를 사용할 수 있도록 업데이트하는 것에 해당한다. 이러한, 업데이트 과정은 액션 컨덕터 모듈(130)을 통해 이루어진다.
- [0059] 6) 만약 입력 포트에 대해서 전송 방향 포트가 하나만 존재한다면, 액션 컨덕터 모듈(130)은 해당 플로우 엔트리 또한 우선순위를 높여 업데이트를 수행한다.
- [0060] 한편, 제일 처음 다수의 플로우 엔트리의 집합체인 플로우 테이블을 전달받은 오픈플로우 스위치(200)는 애플리케이션을 통해 위와 같은 과정을 수행한다. 따라서, 현재 상태에서 보상 값에 기반하여 (Based on Reward Value), 각 포트의 효율성을 검증하고, 최종 전송 경로를 선정하게 된다.
- [0061] 한편, 이러한 전송된 과정에 따른 전송경로 설정을 수행하는 과정에 대해 구성요소에 근거하여 좀 더 상세히 살펴보면 다음과 같다.
- [0062] 어플리케이션 모듈(100)은 데이터 평면에 존재하는 출발지-목적지(Src-Dst) 간의 플로우 테이블(Flow Table)을 미리 전달하도록 구성된다. 이때, 하나의 입력포트(In_port)에 대하여 전송 방향 포트(Output port)가 다수 개일 경우, 전송된 동작을 수행할 수 있다. 여기서, 플로우 테이블은 패킷 전송이 필요할 때마다 요청에 의해 제공되는 것이 아니며, 따라서 패킷 전송 이전에 미리 수신된 플로우 테이블을 이용하여 액션을 수행 (및/또는 학습)하고, 이에 따라 패킷을 전송한다. 이에 따라, 오픈플로우 스위치(200)는 출발지-목적지(Src-Dst) 간 각 전송 경로에 대한 우선순위에 관한 플로우 테이블(Flow Table)을 설정하도록 구성된다.
- [0063] 한편, 어플리케이션 모듈(100)은 상기 플로우 테이블을 기반으로 해당 상태(State)마다 보상값(Reward)이 가장 큰 액션(Action)을 수행하도록 구성된다. 또한, 어플리케이션 모듈(100)은 상기 보상값이 가장 큰 액션을 수행함으로써 패킷을 전송하고, 보상값을 상기 액션에 대응하는 상기 보상값으로 갱신하도록 구성된다. 한편, 상기 패킷은 에지(Edge) 노드에서 전송되는 경우뿐만 아니라, 전체 네트워크 상에서 임의의 노드로부터 전송될 수 있음은 물론이다. 또한, 상기 패킷은 출발지 노드에서 목적지 노드 간의 경로 상의 중간 노드들이 갖고 있는 플로우 테이블을 통해 상기 목적지까지 전송될 수 있다.
- [0064] 상태 모니터 모듈(110)은 입력 포트(In_port)를 제외한 나머지 포트들의 송신/수신 바이트들의 수의 증가 여부와 연관된 상태를 모니터링하도록 구성된다. 구체적으로, 상태 모니터 모듈(110)은 OpenVSwitch가 설치된 하드웨어의 이더넷 포트(Ethernet Port)를 모니터링할 수 있다. 또한, 이러한 모니터링의 경우 지속적으로 동작하는 것이 아니라, 오픈플로우 스위치(200)에서 Packet_IN Message를 수신하는 경우에만 이루어지도록 구성될 수 있다. 즉, 각 OpenVSwitch로 이에 대한 정보를 전달할 때 액션 컨덕터 모듈(130)의 요청에 의해 동작하게 되고, 전송 포트가 결정됨과 동시에 대기상태로 들어간다. 이는 각 노드에서 모니터링에 의한 자원 낭비를 방지하고 Data Plane의 전체적인 부하를 감소시키기 위함이다.
- [0065] 이와 관련하여, 컨트롤러(10)는 패킷이 호스트로부터 오픈플로우 스위치(200)에 유입될 때, 상기 오픈플로우 스위치(200)로부터 상기 패킷을 전송하기 위한 상기 플로우 테이블에 대한 요청 메시지(Packet_IN Message)를 수신한다. 또한, 컨트롤러(10)는 상기 요청 메시지를 전달받아 상기 패킷을 전송할 수 있는 최단 거리 다중 경로를 선정할 수 있다. 이때, 최단 거리 다중 경로는 다중 경로에 대한 플로우 테이블에서 단순히 Hop Count를 활용하여 선정된다. 이후에, 컨트롤러(10)는 최단 거리 다중 경로를 각각의 오픈플로우 스위치(200)로 전달하게 된다.
- [0066] 상기 오픈플로우 스위치(200)는, 각 플로우 테이블에서 상기 전송 방향 포트들을 활용하여 스위치의 현재 상태(State)를 결정할 수 있다. 또한, 상기 어플리케이션 모듈(100)은 보상값 계산모듈(R-Calculator, 130)을 통해 상기 플로우 테이블에서 동일한 입력 포트에 대한 전송 방향 포트 정보를 토대로 보상값 테이블(Reward Table)

을 생성할 수 있다. 일 예로, 보상값 계산모듈(130)은 송신/수신 바이트들에 대하여 각각의 레이트를 계산하여, 상기 레이트의 증가 여부와 연관된 상태를 모니터링하도록 구성될 수 있다. 이에 따라, 보상값 계산모듈(130)은 전송 방향 포트 정보와 레이트 정보에 기반하여 보상값 테이블(Reward Table)을 생성할 수 있다.

[0067] 이와 관련하여, 플로우 테이블 모듈(220)에 해당하는 OpenVSwitch의 상태(State)는 상태 모니터 모듈(110)을 통한 포트 모니터링에 의해 결정된다. 상태 모니터 모듈(110)에서는 수학적 식 1의 (1), (2)를 통해 입력 포트(Input Port)를 제외한 나머지 포트들의 시간 t 및 $t+1$ 에서 Tx bytes, Rx bytes와 관련된 상태를 모니터링하여 액션 컨덕터 모듈(130)로 전달한다.

수학적 식 1

$$rxRate(t) = rxBytes(t) - rxBytes(t-1) \quad (1)$$

$$txRate(t) = txBytes(t) - txBytes(t-1) \quad (2)$$

[0068]

[0069] 여기서, $rxRate(t)$ 는 시간 t 와 $t-1$ 사이의 패킷 수신량에 해당하고, $txRate(t)$ 는 시간 t 와 $t-1$ 사이의 패킷 송신량에 해당한다. 또한, $rxBytes(t)$ 와 $rxBytes(t-1)$ 는 시간 t 와 $t-1$ 에서 각 패킷 수신량에 해당한다. 또한, $txBytes(t)$ 와 $txBytes(t-1)$ 는 시간 t 와 $t-1$ 에서 각 패킷 송신량에 해당한다.

[0070] 구체적으로, 시간 t 및 $t+1$ 에서의 Tx bytes, Rx bytes가 동일할 경우 0(미사용중), 증가했을 경우 1(사용중)로 나타낼 수 있다. 즉, 송신/수신 바이트들의 수는 시간 t 또는 $t+1$ 까지의 송신/수신된 누적 바이트들의 수에 해당하고, 송신/수신 바이트들의 수와 동일하다는 의미는 시간 t 에서 $t+1$ 로 변할 때 새로 송신/수신되는 데이터가 없다는 의미이다. 한편, 상태 모니터 모듈(110)은 상기 시간 $t+1$ 에서 송신/수신 바이트들의 수가 상기 시간 t 에서의 송신/수신 바이트들의 수보다 증가하면, 상기 해당 상태를 1로 정의한다. 즉, 송신/수신 바이트들의 수는 시간 t 또는 $t+1$ 까지의 송신/수신된 누적 바이트들의 수에 해당하고, 송신/수신 바이트들의 수가 증가한다는 의미는 시간 t 에서 $t+1$ 로 변할 때 새롭게 송신/수신되는 데이터가 존재한다는 의미이다. 이때, 출력 포트(Output Port)들이 얻는 0 또는 1값의 집합을 상태(State)로 정의한 후, 요청이 있을 시 모니터링하여 액션 컨덕터 모듈(130)로 전달하게 된다.

[0071] 이와 같이, 새롭게 송신/수신되는 데이터의 존재 유무 또는 다른 기준에 따라 해당 상태를 다양하게 정의하는 것이 가능하면, 이러한 해당 상태를 정의하는 방법은 이에 한정되는 것이 아니라 자유롭게 변형 가능하다.

[0072] 또한, 액션 컨덕터 모듈(Action conductor, 130)은 송신/수신 바이트들의 수의 증가 여부와 연관된 상태를 전달받아 해당 상태(State)마다 보상값(Reward)이 가장 큰 액션(Action)을 수행하도록 구성된다. 또한, 액션 컨덕터 모듈(130)은 상기 송신/수신 바이트들의 수의 증가 여부 및 상기 각각의 레이트의 증가 여부와 연관된 상태를 전달받아 해당 상태(State)마다 보상값이 가장 큰 액션(Action)을 수행하도록 구성될 수 있다. 이때, 상기 송신/수신 바이트들의 수의 증가 여부 및 상기 각각의 레이트의 증가 여부는 각각 상태 모니터 모듈(110)과 보상값 계산모듈(120)로부터 모니터링되어 이와 관련된 상태가 액션 컨덕터 모듈(130)로 전달될 수 있다.

[0073] 여기서, 상기 보상값(Reward)은 상기 액션 컨덕터 모듈(Action conductor, 120)이 상기 액션을 수행하였을 때 해당하는 포트에서 측정되는 전송 지연(Transmission Delay) 값의 역수로 계산될 수 있다.

[0074] 이와 관련하여, 도 5는 본 발명에 따른 Application (또는 어플리케이션 모듈)이 OpenVSwitch(오픈플로우 스위치)로부터 얻을 수 있는 각 상태를 나타낸 도면이다. 예를 들어, 도 5에 도시된 바와 같이, $s(1)$ 내지 $s(8)$ 의 8가지 상태에 대하여, Application (또는 어플리케이션 모듈(100))이 OpenVSwitch(오픈플로우 스위치(200))로부터 얻을 수 있는 각 상태(State)를 나타낸 것이다.

[0075] 한편, 액션 컨덕터 모듈(130)에서는 앞서 정의된 상태(State), 액션(Action), 보상값(Reward)를 통해 각 State

와 Action에 대한 Reward 테이블을 생성한다. 도 3과 관련되어 전술된 SARSA 알고리즘을 이용하는 경우, 에피소드가 끝나는 시점에서 해당 Q 테이블을 학습한 후, 이를 정책(Policy)으로 사용한다. 하지만, 도 4 및 도 5에서 제안하는 방식에서는 State가 포트의 사용 유무에 의해 정의되어 있기 때문에 실제 같은 State 이더라도 Action으로부터의 Reward는 달라질 수 있다. 이와 같은 이유로 지속적인 Q 테이블 업데이트를 통해 효율적인 Action을 선택할 수 있도록 하는 알고리즘 적용이 가능하다.

[0076] 구체적으로, 이러한 알고리즘은 OpenVSwitch에서 전송 포트를 선정하기 위한 알고리즘이다. 또한, 기존 강화학습 알고리즘과 다른 점은, ϵ -greedy한 학습 정책이 아니라는 점이다. 제안하는 방식에서는 상태(State)가 각 포트의 사용 유무에 의해 정의되어 있기 때문에 실제 같은 상태(State)이더라도 행동(Action)으로부터의 보상값(Reward)은 각 Packet_IN 이벤트마다 달라질 수 있다. 이와 같은 이유로 패킷이 유입되는 이벤트를 감지하고 그때의 상태(State)와 행동(Action)을 통해 전송 포트를 선정할 수 있도록 적용할 수 있다.

[0077] 이와 관련하여, 표 2는 도 4 및 도 5에서 제안하는 방식에 따라 적용된 알고리즘의 의사코드(pseudo code)를 나타낸 것이다.

표 2

Algorithm	Port Decision on OpenFlow Switch
1:	<i>Initialize $Q(s, a)$ arbitrarily</i>
2:	<i>if Packet inflow is detected</i>
3:	<i>Choose a from s Randomly</i>
4:	<i>loop</i>
5:	<i>Take Action a</i>
6:	<i>Observe Reward r and Choose a' from s</i>
7:	<i>Update $R(s, a)$ in Table</i>
8:	<i>$a \leftarrow a'$</i>
9:	<i>until Reward table complete</i>
10:	<i>Choose a which has high Reward value</i>

[0078]

[0079] 즉, 액션 컨덕터 모듈(130)은 상기 보상값에 대한 보상값 테이블 $Q(s,a)$ 에서, 패킷 유입이 검출되면 상태(s)로부터 제1액션(a)을 임의로 선택하고, 상기 제1액션(a)을 수행하여, 상기 보상값(r)을 관측한다. 또한, 액션 컨덕터 모듈(130)은 상기 상태(s)로부터 제2액션(a')을 선택하여 상기 보상값 테이블 $Q(s,a)$ 를 $Q(s, a')$ 으로 업데이트한다. 또한, 액션 컨덕터 모듈(130)은 상기 보상값 테이블 중 상기 상태(s)와 연관된 보상값 테이블(reward table)이 완성(complete)될 때까지 상기 보상값 관측 및 상기 보상값 테이블 업데이트를 반복한다. 이러한 상기 반복된 보상값 관측을 통해 상기 보상값이 가장 큰 액션을 선택하여, 경로 결정(Path decision)을 수행할 수 있다. 즉, 액션 컨덕터 모듈(130)은 상기 포트들의 사용 유무와 상기 수신된 플로우 테이블에 기반하여, 상기 해당 상태(State)마다 상기 보상값(Reward)이 가장 큰 액션(Action)을 수행하도록 구성될 수 있다. 이때, 상기 포트들의 사용 유무에 의해 상기 상태가 정의되어, 상기 상태에 대한 액션으로부터의 보상값은 상기 포트들의 사용 유무에 따라 달라질 수 있다. 한편, 입력포트에 대하여 전송 방향 포트가 하나만 존재한다면, 해당 플로우 엔트리에 대한 우선순위를 높여 업데이트를 수행할 수 있다.

[0080] 본 발명에서 제안하는 Application을 통해 패킷이 전송되는 구체적인 시나리오는 다음과 같다. Packet_IN 메시지에 의해 계산되는 전송 경로는 각 OpenVSwitch로 전달된다. 이때, 플로우 엔트리(Flow Entry)의 형태는 In 포트와 Output 포트가 한 쌍이 되어 하나의 전송 링크에 대해 총 2가지의 플로우 엔트리를 전달받게 된다. 이때, 효율적인 전송 포트를 찾기 위하여 각 매치(Match) 필드 상의 In 포트 정보를 기반으로 각각의 보상값 테이블(Reward table)을 만들게 된다. 오픈플로우 스위치(200)에서는 출발지-목적지 간 전송 경로를 홑 수 기반의 알고리즘으로 계산하고 각 전송 경로에 대한 우선순위를 다르게 설정하여 OpenVSwitch에 배치한다. 여기서, 전술된 출발지-목적지 간 전송 경로를 홑 수 기반의 알고리즘과 관련하여, 데이크스트라(Dijkstra) 알고리즘이 사용될 수 있다. 즉, 데이크스트라 알고리즘은 방향이 주어진 가중 그래프(weighted graph) G와 출발점 s를 입력으

로 받는다. 그래프 G의 모든 꼭짓점들의 집합을 V라 하고, 그래프의 변을 출발점 u와 도착점 v의 순서쌍 (u, v)로 표현한다. G의 모든 변들의 집합을 E라 하고, 변들의 가중치는 함수 $w: E \rightarrow [0, \infty]$ 로 표현한다. 이때 가중치 $w(u, v)$ 는 꼭짓점 u에서 꼭짓점 v로 이동하는 데 드는 비용(시간, 거리 등)이 된다. 경로의 비용은 경로 사이의 모든 변들의 가중치의 합이 된다. 데이크스트라 알고리즘은 V의 임의의 꼭짓점의 쌍 s와 t가 있을 때 s에서 t로 가는 가장 적은 비용이 드는 경로(최단 경로)를 찾는데 활용될 수 있다.

[0081] 한편, 아래의 표 3은 하나의 OpenVSwitch에 배치되는 출발지-목적지 간의 Flow Entry에 대한 Table의 예시를 나타낸다.

표 3

Flow Entry	Match Field	Action
Flow 1	priority=3, in_port=1, dst_mac, src_mac, dst_ip, src_ip	output:2
Flow 2	priority=3, in_port=2, dst_mac, src_mac, dst_ip, src_ip	output:1
Flow 3	priority=2, in_port=1, dst_mac, src_mac, dst_ip, src_ip	output:3
Flow 4	priority=2, in_port=3, dst_mac, src_mac, dst_ip, src_ip	output:1
Flow 5	priority=1, in_port=1, dst_mac, src_mac, dst_ip, src_ip	output:4
Flow 6	priority=1, in_port=4, dst_mac, src_mac, dst_ip, src_ip	output:1

[0082]

[0083] 즉, 상기 플로우 테이블은, 플로우 엔트리(Flow Entry), 매치 필드(Match Field) 및 액션(Action) 필드를 포함할 수 있다. 여기서, 플로우 엔트리는 복수의 플로우들을 서로 구분하도록 구성된다. 또한, 매치 필드는 상기 플로우 엔트리에 대하여, 상기 각 전송 경로에 대한 우선순위, 입력 포트(in_port), 상기 출발지 및 상기 목적지의 MAC 주소(src_mac, dst_mac)와 IP 주소(dst_ip, src_ip)를 포함한다. 또한, 액션 필드는 상기 플로우 엔트리에 대하여, 상기 입력 포트(in_port)에 대응하는 출력 포트(output port)를 포함한다.

[0084] 한편, 액션 컨덕터 모듈(130)은 상기 플로우 테이블 내의 각 플로우 엔트리의 입력 포트(In port) 정보를 추출하고, 상기 입력 포트에 대응하는 적어도 하나의 출력 포트(output port)의 가용 대역폭을 기반으로 상기 보상값 계산모듈(120)로부터 상기 보상값을 전달받아 최종 액션을 결정하도록 구성될 수 있다.

[0085] 구체적으로, 액션 컨덕터 모듈(130)에서는 배치되어 있는 플로우 테이블 내에서 각 플로우 테이블의 In 포트 정보를 추출한다. 이는 In 포트에 대한 행동(Action) Rule의 정보를 기반으로 Reward Table을 만들기 위해 수행되는 과정이다. 구체적으로, 표 3을 참조하면, in_port=1, in_port=2, in_port=3, in_port=4에 대한 Table이 만들어진다. 이때, in_port=1에 대한 Table은 행동(Action) Rule을 output:2, output:3 및 output:4를 갖고 있으므로 현재의 상태(State)에서 in_port=1에 대한 각각의 Output 포트의 가용대역폭을 기반으로 보상값 계산모듈(R Calculator)(120)로부터 보상값(Reward)을 전달받아 최종 행동(Action)을 결정하게 된다. 또한, in_port=2, in_port=3, in_port=4에 대한 Table의 경우, output:1에 해당하는 하나의 행동(Action)만 갖게 되므로 해당 Flow Entry를 선택하게 된다. 각 Reward Table에서 보상값(Reward) 값에 의해 행동(Action)을 결정 한 후, 어플리케이션 모듈(110)은 해당 행동(Action)을 갖는 Flow Entry의 우선순위를 높게 적용하여 업데이트할 수 있다. 또한, 각 Entry의 경우 Timeout 시간을 적용하여 해당 Entry가 활용되지 않을 경우 상기 Timeout 시간 후에 삭제되도록 할 수 있다. 여기서, 상기 Timeout 시간은 5초로 결정될 수 있다.

[0086] 한편, 도 6은 본 발명의 일 실시예에 따른 각각의 입력 포트에서의 보상값 테이블을 도시한다. 도 6에 도시되고 진술된 바와 같이, 각각의 입력 포트에서의 출력 포트는 적어도 하나 이상이고, 예를 들어, in_port=1에 대한 액션 규칙은 복수의 출력 포트(output:2, output:3 및 output:4)에 대해서 서로 다르게 정의될 수 있다. 즉, 각 행동(Action)에 대한 보상값(Reward)을 기반으로 최종적으로 플로우 엔트리(Flow Entry)를 결정하게 된다. 이렇게 Data Plane의 각 OpenVSwitch에서 최종 플로우 엔트리를 업데이트 하게 되면 해당 Entry들을 기반으로 만들

어진 경로를 통해 패킷이 전송된다.

[0087] in_port=2, in_port=3, in_port=4의 Reward Table을 통해 결정된 Flow Entry의 경우, 이웃 노드로부터의 패킷 전송 여부에 따라 활용되거나 삭제된다. 이 의미는 2번 포트와 연결된 이웃 노드로부터 패킷이 전달된다면 in_port=3과 in_port=4를 매치 필드(Match Field)로 갖는 플로우 엔트리(Flow Entry)는 Timeout 설정에 의해 삭제되고 in_port=2를 Match 필드로 갖는 플로우 엔트리만 활용되어 패킷을 목적지까지 전달하게 된다. 본 방식은 오픈플로우 스위치가 선정한 출발지에서 목적지까지의 Hop수 기반 다중 경로에 대하여 OpenVSwitch의 Application을 통해 현재 상태(State)에서 가장 효율적인 전송경로를 스스로 결정하게 된다. 이러한 방식을 제안하게 된 가장 큰 이유는 SDN 환경에서 Data Plane의 자원을 적극 활용하여 보다 유연한 전송 경로 설정 방법을 탐색하기 위함이다.

[0088] 전송된 보상값(Reward)의 경우, 각 플로우 엔트리의 In 포트의 정보를 토대로 Output 포트를 추출한다. 또한, 각 Output 포트에 대한 보상값(Reward) 계산을 수행하게 될 때, 해당 포트의 가용 대역폭을 모니터링하여 계산한 후 액션 컨텍터 모듈(130)로 전달한다. 전송된 수학적 식 1의 (1)과 (2)를 통해 이더넷 포트의 패킷 송수신 여부, 즉 포트의 사용량을 확인할 수 있다. 이때, 액션 컨텍터 모듈(130)의 요청에 의해 SMonitor 모듈이 송수신 바이트들(Bytes)을 모니터링 한다. 이때 송신 및 수신된 바이트들이 증가했다면 현재 포트가 사용되고 있다고 판단하여 상태(State) 결정에 반영한다. 또한, 각 Rate가 증가하였을 경우 이를 가용대역폭과 전송 지연 계산에 활용하여 보상값(Reward)을 계산한다.

[0089] 이와 관련하여, 수학적 식 2의 (3) 내지 (5)는 Application이 해당 상태(State)에서 각 Reward Table에 대하여 행동(Action)을 수행했을 때의 보상값(Reward) 값을 계산하기 위한 수식이다. 이는 각 포트의 최대 대역폭을 활용하여 현재 활용할 수 있는 가용대역폭을 구하고 이를 기반으로 전송 지연 값을 계산하게 된다. 수학적 식 2의 (3)과 (4)에서 8을 곱해주는 이유는 대역폭의 단위가 Bit per Second 이기 때문에 Byte 단위로 얻는 Rate 값을 Bit로 바꿔주기 위함이다.

수학적 식 2

$$AvailableBW(t) = [MaxBW - (txRate(t) + rxRate(t)) * 8] \quad (3)$$

$$TDelay(t) = \frac{(PLength * 8)}{AvailableBW(t)} \quad (4)$$

$$Reward(t) = \frac{1}{TDelay(t)} \times Weight \quad (5)$$

[0090]

[0091] 여기서, $AvailableBW(t)$ 와 $MaxBW(t)$ 는 각각 시간 t에서 포트의 가용 대역폭과 포트의 최대 대역폭을 나타낸다. 또한, $TDelay(t)$ 과 $PLength$ 는 각각 시간 t에서 포트의 전송 지연 값과 유입된 패킷(Packet)의 크기(Size)를 나타낸다. 또한, $Reward(t)$ 는 시간 t에서 해당 포트의 보상값을 나타낸다.

[0092] 수식 (3)과 (4)는 Application이 해당 상태(State)에서 각 Reward Table에 대하여 행동(Action)을 수행했을 때의 보상값(Reward) 값을 계산하기 위한 수식이다. 이는 각 포트의 최대 대역폭을 활용하여 현재 활용할 수 있는 가용대역폭을 구하고 이를 기반으로 전송 지연 값을 계산하게 된다. 수식 (3)과 (4)에서 8을 곱해주는 이유는 대역폭의 단위가 Bit per Second 이기 때문에 Byte 단위로 얻는 Rate 값을 Bit로 바꿔주기 위함이다.

[0093] 보상값(Reward) 값은 전송 지연 값의 역수로 정의되며 이는 각 포트의 가용대역폭이 같거나 다를 경우 패킷 전송을 수행함으로써 얻을 수 있다. 또한 가중치(Weight)의 경우, 최대 대역폭의 값에 따라 결정되는데, 이는 보상값(Reward)이 커지는 것을 방지하고 적정 수준으로 각 포트를 비교하기 위해 곱해준다. 이러한 과정을 통해 얻게 되는 전송 지연값을 기반으로 Application은 각 행동(Action)을 수행하고 Output에 해당하는 포트 중, 보

상값(Reward) 값이 가장 큰 Port를 선정하여 패킷을 전송하게 된다.

- [0094] 이와 관련하여, 수학적 2에 따르면, 보상값(Reward)이 액션 컨덕터 모듈(Action conductor, 120)이 액션(Action)을 수행하였을 때, 해당 포트(Port)에서 측정되는 전송 지연 값의 역수로 나타남을 알 수 있다. 한편, 어플리케이션 모듈(100)은, 상기 보상값이 가장 큰 액션을 수행함으로써 패킷을 전송하고, 상기 액션에 대응하는 상기 보상값을 갱신하도록 구성될 수 있다. 즉, 패킷 전송에 따라 해당 포트에서 측정되는 전송 지연이 최소가 되는 경로를 선택하도록 구성함으로써 보상값을 최대가 될 수 있다.
- [0095] 도 7은 본 발명의 일 실시예에 따라, Packet_IN 메시지를 기반으로 컨트롤러가 OpenVSwitch에 플로우 테이블을 배치한 것을 나타낸다. 컨트롤러에서 선정된 다중 경로와 관련하여, 총 5개의 경로가 생성될 수 있다. 즉, SW1-SW2-SW3-SW4를 지나는 경로 A와 SW1-SW4-SW5-SW6을 지나는 경로 B, SW1-SW5-SW6을 지나는 경로 C, SW1-SW2-SW6을 지나는 경로 D, 마지막으로 SW1-SW2-SW5-SW6을 지나는 경로 E까지 총 5개의 경로가 생성되며 이에 대한 Flow Table을 경로상의 OpenVSwitch에게 전달된다. 각 Entry는 컨트롤러가 출발지에서 목적지까지의 다중 경로를 선정한 후 배치한 것으로서 SW1, SW2, SW5, SW6에는 하나의 In 포트에 대하여 여러 개의 Action Rule이 생성된다. 이는 현재 OpenVSwitch의 상태(State)에서 유입되는 패킷을 전송하기 위해 수행할 수 있는 행동(Action)이며 이를 기반으로 각 Entry의 Output Port를 기반으로 패킷을 전송함으로써 보상값(Reward) 값을 얻게 된다.
- [0096] 각 Flow Entry는 서로 다른 우선순위를 갖고 있다. 이는 동일한 Match 필드에서 다양한 Action 필드를 갖게 하는 역할을 하며 최종 Action이 결정되면 우선순위를 다른 Entry보다 높게 설정하여 패킷이 Match 필드에 있는 In 포트에 유입될 경우 활용하게 된다.
- [0097] 한편, 도 8은 본 발명의 일 실시예에 따라, 전송 경로가 설정된 형태를 나타낸다. 도 8에 도시된 바와 같이, SW1과 SW2에서 수행할 수 있는 각 행동(Action)중에 보상값(Reward) 값이 가장 큰 Flow Entry를 선택한 것이다. 또한, 각 OpenVSwitch에서 활용되지 않는 Flow Entry의 경우 Timeout 설정에 의해 삭제되어 현재 패킷 전송에 활용되고 있는 Flow Entry만 남아있다. 경로 선정 이후, 또 다른 Host에 의한 패킷 전송이 있을 시, 위에서 언급한 절차를 거쳐 상태(State)를 확인하고 수행할 수 있는 행동(Action)에 따른 보상값(Reward) 값을 얻음으로써 경로가 선정된다. 본 발명에서는 가장 효율적인 패킷 전송 경로 설정에 대하여 오픈플로우 스위치의 추가 개입 없이 Data Plane의 OpenVSwitch가 각 포트의 현재 상태 및 효율성을 확인하여 전송링크가 선정된다. 따라서, 컨트롤러의 추가 개입 없이 전송링크가 선정되기 때문에 Self Routing Organization 기법이라 명시할 수 있다.
- [0098] 이상에서는 본 발명에 따른 소프트웨어 정의 네트워크에서 데이터 평면 애플리케이션을 활용하여 전송 경로를 설정하는 전송 경로 설정 장치에 대해 살펴보았다. 아래에서는, 본 발명의 다른 양상에 따른 소프트웨어 정의 네트워크에서 데이터 평면 애플리케이션을 활용하여 전송 경로를 설정하는 전송 경로 설정 방법에 대해 살펴보기로 한다. 한편, 진술된 전송 경로 설정 장치에 대한 내용은 전송 경로 설정 방법에도 활용될 수 있음은 물론이다.
- [0099] 이와 관련하여, 도 9는 본 발명에 따른 소프트웨어 정의 네트워크에서 데이터 평면 애플리케이션을 활용하여 전송 경로를 설정하는 전송 경로 설정 방법의 흐름도를 도시한다. 전송 경로 설정 방법은 전송 경로 설정 장치에 의해 수행되며, 특히, 어플리케이션 모듈(100) 및 오픈플로우 스위치(200) 중 적어도 하나에 의해 수행될 수 있다.
- [0100] 도 9에 도시된 바와 같이, 전송 경로 설정 방법은 상태 모니터링 단계(S100), 레이트 모니터링 단계(S150) 및 액션 수행 단계(S200)를 포함한다. 한편, 액션 수행 단계(S200)는 제1액션 선택 단계(S210), 보상값 관측 단계(S220), 및 제2액션 선택/보상값 테이블 업데이트 단계(S230)를 포함한다. 또한, 액션 수행 단계(S200)는 보상값 테이블 완성 여부 판단 단계(S240) 및 경로 결정 단계(S250)를 더 포함할 수 있다.
- [0101] 상태 모니터링 단계(S100)에서, 입력 포트(In_port)를 제외한 나머지 포트들의 송신/수신 바이트들의 수의 증가 여부와 연관된 상태를 모니터링한다.
- [0102] 레이트 모니터링 단계(S150)에서, 상기 송신/수신 바이트들에 대하여 각각의 레이트를 계산하여, 상기 레이트의 증가 여부와 연관된 상태를 모니터링한다.
- [0103] 한편, 상태 모니터링 단계(S100)에서, 시간 t+1에서 송신/수신 바이트들의 수가 시간 t에서의 송신/수신 바이트들의 수와 동일하면 상기 해당 상태를 0으로 정의한다. 또한, 상기 시간 t+1에서 송신/수신 바이트들의 수가 상기 시간 t에서의 송신/수신 바이트들의 수보다 증가하면, 상기 해당 상태를 1로 정의하여, 상기 포트들의 상태

를 모니터링한다.

- [0104] 액션 수행 단계(S200)에서, 송신/수신 바이트들의 수의 증가 여부와 연관된 상태를 전달받아 해당 상태(State)마다 보상값(Reward)이 가장 큰 액션(Action)을 수행한다.
- [0105] 구체적으로, 제1액션 선택 단계(S210)에서, 상기 보상값에 대한 보상값 테이블 $Q(s,a)$ 에서, 패킷 유입이 검출되면 상태(s)로부터 제1액션(a)을 임의로 선택한다.
- [0106] 또한, 보상값 관측 단계(S220)에서, 상기 제1액션(a)을 수행하여, 상기 보상값(r)을 관측한다. 이에 따라, 제2액션 선택/보상값 테이블 업데이트 단계(S230)에서, 상기 상태(s)로부터 제2액션(a')을 선택하여 상기 보상값 테이블 $Q(s,a)$ 를 $Q(s, a')$ 으로 업데이트한다.
- [0107] 다음으로, 보상값 테이블 완성 여부 판단 단계(S240)에서, 상기 보상값 테이블 중 상기 상태(s)와 연관된 보상값 테이블(reward table)의 완성(complete) 여부를 판단한다. 이때, 상기 보상값 테이블이 완성될 때까지 상기 보상값 관측 단계(S220) 및 상기 제2액션 선택/보상값 테이블 업데이트 단계(S230)를 반복한다.
- [0108] 상기 상태(s)와 연관된 보상값 테이블(reward table)이 완성된 것으로 판단되면, 경로 결정 단계(S250)에서, 상기 반복된 보상값 관측을 통해 상기 보상값이 가장 큰 액션을 선택하여, 경로 결정(Path decision)을 수행할 수 있다.
- [0109] 본 발명에 따르면, 경로설정을 제외한 다른 기능을 수행해야 하는 컨트롤러를 통해 운용되는 네트워크상에서는 본 발명에서 제안하는 방식을 통해 컨트롤러의 부하를 줄일 수 있다. 또한, 많은 노드들을 포함하는 대규모 네트워크에서도 Flow Table 요청에 따른 부하를 줄일 수 있다.
- [0110] 본 발명의 기대효과와 관련하여, 본 발명에서 제안된 방식을 통해 전송경로 설정 기능을 보조함으로써 컨트롤러의 부하를 줄이고 컨트롤러가 해야 하는 다른 기능을 원활하게 수행할 수 있는 환경을 제공할 수 있다.
- [0111] 본 발명의 사업화 전망과 관련하여, 소프트웨어 정의 네트워크를 활용하는 통신망에 본 발명에서 제안한 방식을 적용한다면 컨트롤러의 부하를 줄임으로써 사용자들이 보다 원활한 네트워크 환경을 제공받을 수 있을 것으로 기대된다.
- [0112] 본 발명의 적어도 일 실시예에 따른 전송 경로 설정 방법은, 소프트웨어 정의 네트워크에서 패킷전송만을 담당했던 데이터 평면에 전송 경로 갱신 및 요청 등의 기능을 Application 형태로 배치하여 처리함으로써, 컨트롤러에 집중되는 부하를 감소시킬 수 있다는 장점이 있다.
- [0113] 또한, 본 발명의 적어도 일 실시예에 따른 전송 경로 설정 방법은, 소프트웨어 정의 네트워크에서 패킷전송만을 담당했던 데이터 평면에 전송 경로 갱신 및 요청 등의 기능을 Application 형태로 배치하여 처리함으로써, 컨트롤러에 집중되는 부하에 따라 사용자 데이터의 전송 지연이 증가하는 것을 감소시킬 수 있다는 장점이 있다.
- [0114] 소프트웨어적인 구현에 의하면, 본 명세서에서 설명되는 절차 및 기능뿐만 아니라 각각의 구성 요소들은 별도의 소프트웨어 모듈로도 구현될 수 있다. 상기 소프트웨어 모듈들 각각은 본 명세서에서 설명되는 하나 이상의 기능 및 작동을 수행할 수 있다. 적절한 프로그램 언어로 쓰여진 소프트웨어 어플리케이션으로 소프트웨어 코드가 구현될 수 있다. 상기 소프트웨어 코드는 메모리에 저장되고, 제어부(controller) 또는 프로세서(processor) (또는 어플리케이션 프로세서)에 의해 실행될 수 있다.

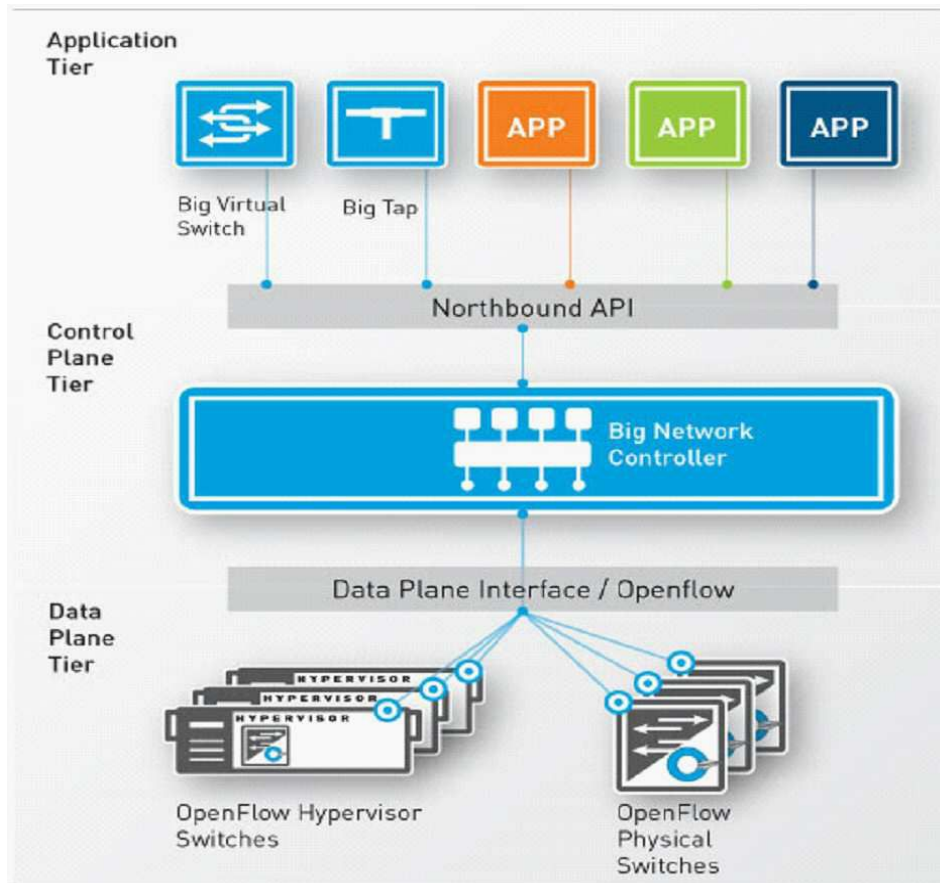
부호의 설명

- [0115] 10: 컨트롤러
- 100: 어플리케이션 모듈
- 110: 상태 모니터 모듈
- 120: 보상값 계산모듈
- 130: 액션 컨딕터 모듈
- 200: 오픈플로우 스위치
- 210: 포트 상태 제공 모듈
- 220: 플로우 테이블 모듈

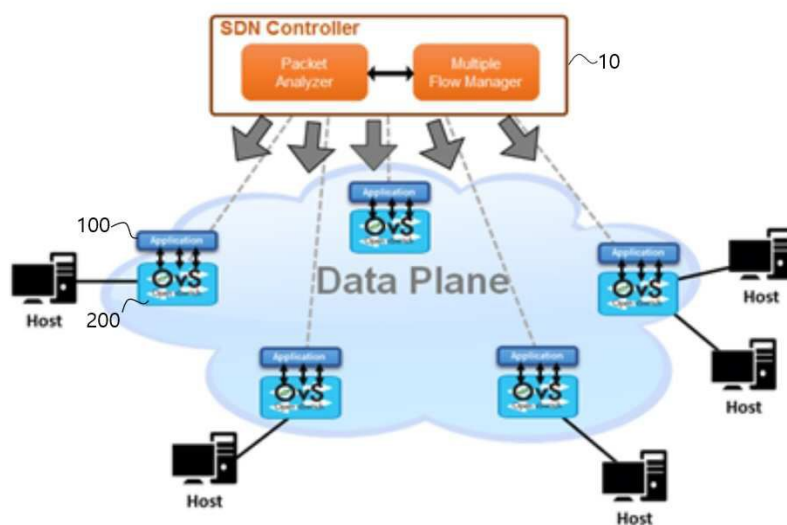
1000: 전송 경로 설정 장치

도면

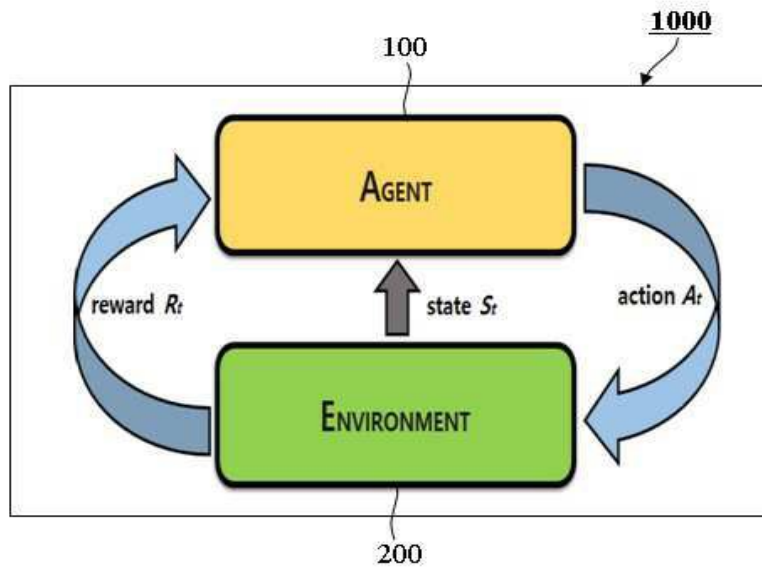
도면1



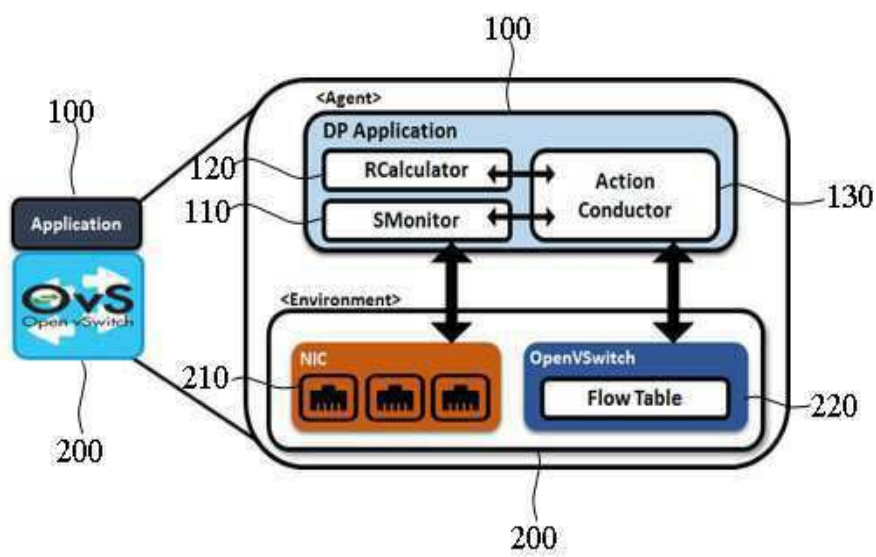
도면2



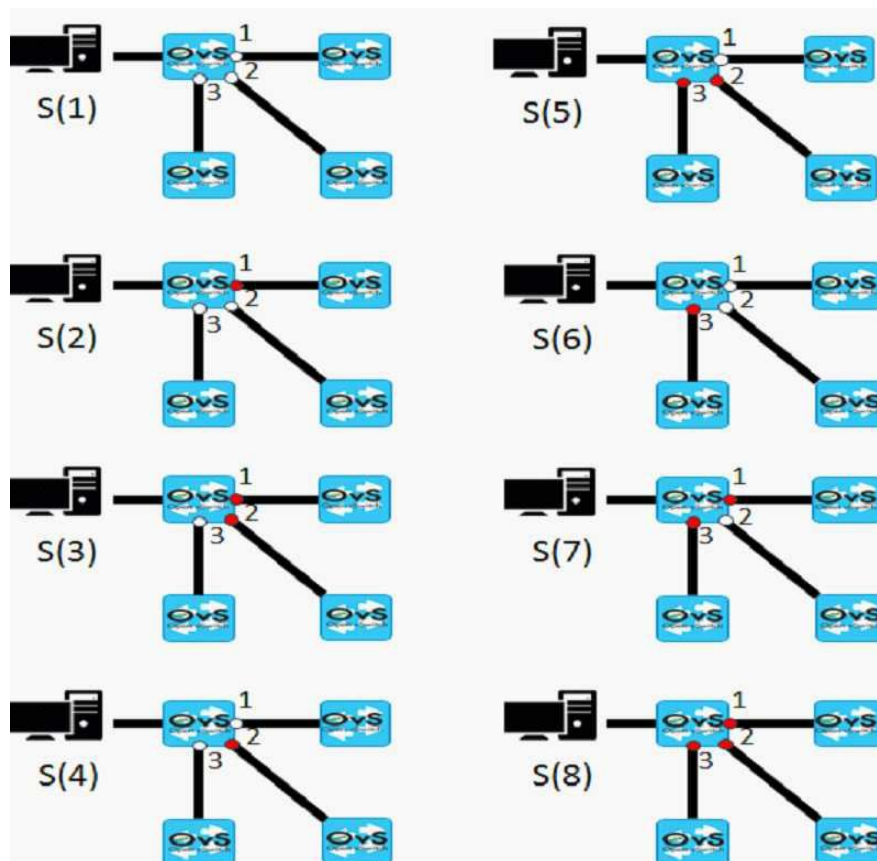
도면3



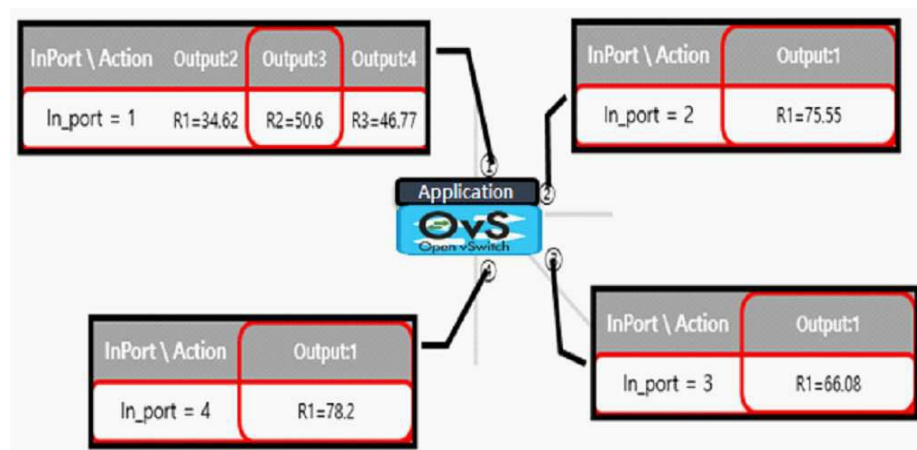
도면4



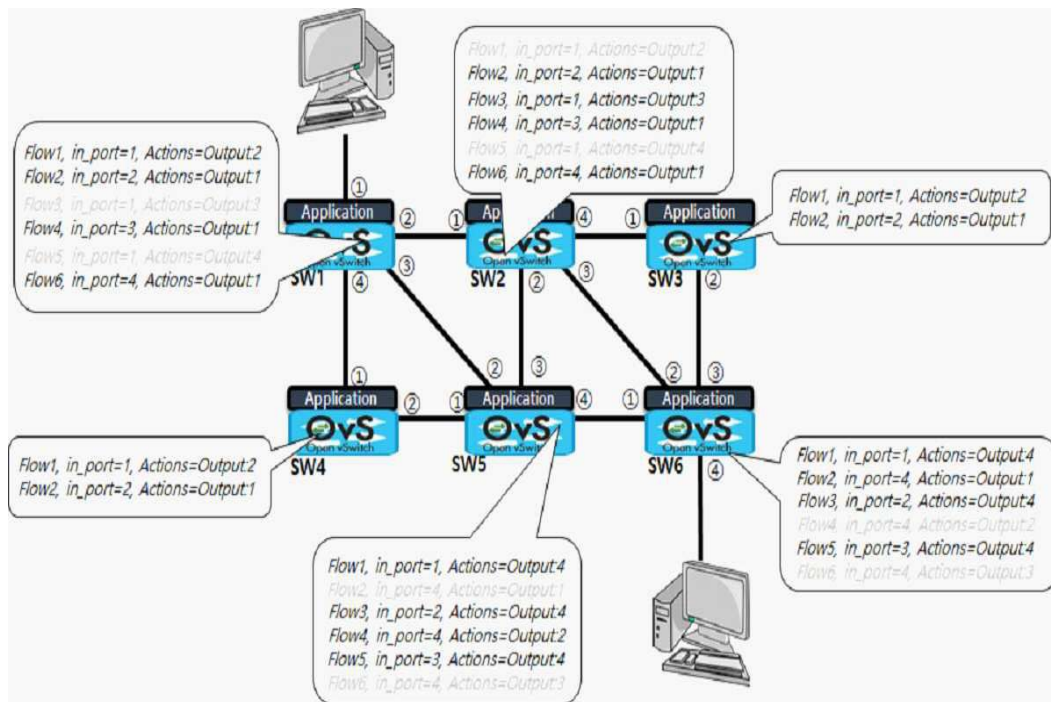
도면5



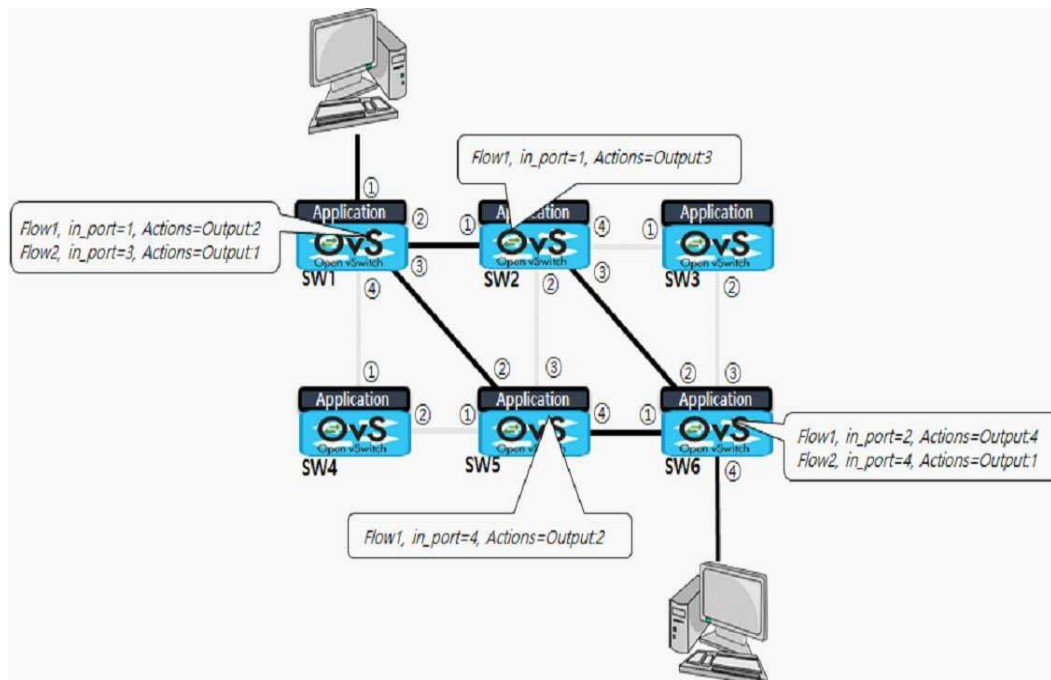
도면6



도면7



도면8



도면9

